

# Intermediate R: Statistical Analysis

Lisa Federer, Research Data Informationist

March 28, 2016

This course is designed to provide an overview of R's `ggplot2` package, which can help create highly customized visualizations. This handout will walk you through every step of today's class. Throughout the handout, you'll see the example code displayed like this:

```
> print(2 + 2)  
  
[1] 4
```

The part that is in *italics* and preceded by the `>` symbol is the code that you will type. The part below it is the output you should expect to see. Sometimes code doesn't have any output to print; in that case, you'll just see the code and nothing else.

Also, sometimes the code is too long to fit on a single line of text in this handout. When that is the case, you will see the code split into separate lines, each starting with a `+`, like this:

```
> long_line_of_code <- c("Some really long code", "oh my gosh, how long is it going to be?",  
+   "is it going to go on forever?", "I don't know, AGGGHHHHH",  
+   "please, make it stop!")
```

When this is the case, do not insert any line breaks, extra spaces, or the plus sign - your code should be typed as one single line of code. Note that the default for your display in R Studio is not to wrap lines of text in your code, but you can turn this on by going to Tools > Global Options > Code Editing, and check the box next to "Soft-wrap R source files."

## 1 Getting Started

Set your working directory and read in the data file, which was originally downloaded from Carnegie Mellon University, but as of March 2016, this URL no longer works. Here, I've specified the folder on my computer where my files are, but you would use the path where your data files are stored.

```
> setwd("Z:/Data Services Workgroup/Data Instruction/R Classes/Statistical Analysis")  
> dat <- read.csv(file = "Plasma_Retinol.csv")
```

Since we're just using this one data frame, we can use a shortcut to save ourselves some type and typing - the `attach` function. This allows us to refer to variables by name, instead of having to specify the data frame name first.

```
> attach(dat)
```

We need several packages for what we'll be doing. Let's load these packages before we begin.

```
> library(psych)  
> library(ggplot2)  
> library(QuantPsyc)  
> library(pastecs)
```

## 2 Descriptive and Summary Statistics

The `summary()` function provides some very basic summary statistics about the data. Depending on the data type of each variable (ie integer, number, factor, etc), different statistics will be provided.

```
> summary(dat)
```

age		sex		smokstat		quetelet	
Min.	:19.00	Min.	:1.000	Min.	:1.000	Min.	:16.33
1st Qu.	:39.00	1st Qu.	:2.000	1st Qu.	:1.000	1st Qu.	:21.80
Median	:48.00	Median	:2.000	Median	:2.000	Median	:24.74
Mean	:50.15	Mean	:1.867	Mean	:1.638	Mean	:26.16
3rd Qu.	:62.50	3rd Qu.	:2.000	3rd Qu.	:2.000	3rd Qu.	:28.85
Max.	:83.00	Max.	:2.000	Max.	:3.000	Max.	:50.40
vituse		calories		fat		fiber	
Min.	:1.000	Min.	: 445.2	Min.	: 14.40	Min.	: 3.10
1st Qu.	:1.000	1st Qu.	:1338.0	1st Qu.	: 53.95	1st Qu.	: 9.15
Median	:2.000	Median	:1666.8	Median	: 72.90	Median	:12.10
Mean	:1.965	Mean	:1796.7	Mean	: 77.03	Mean	:12.79
3rd Qu.	:3.000	3rd Qu.	:2100.4	3rd Qu.	: 95.25	3rd Qu.	:15.60
Max.	:3.000	Max.	:6662.2	Max.	:235.90	Max.	:36.80
alcohol		cholesterol		betadiet		retdiet	
Min.	: 0.000	Min.	: 37.7	Min.	: 214	Min.	: 30.0
1st Qu.	: 0.000	1st Qu.	:155.0	1st Qu.	:1116	1st Qu.	: 480.0
Median	: 0.300	Median	:206.3	Median	:1802	Median	: 707.0
Mean	: 3.279	Mean	:242.5	Mean	:2186	Mean	: 832.7
3rd Qu.	: 3.200	3rd Qu.	:308.9	3rd Qu.	:2836	3rd Qu.	:1037.0
Max.	:203.000	Max.	:900.7	Max.	:9642	Max.	:6901.0
betaplasma		retplasma					
Min.	: 0.0	Min.	: 179.0				
1st Qu.	: 90.0	1st Qu.	: 466.0				
Median	: 140.0	Median	: 566.0				
Mean	: 189.9	Mean	: 602.8				
3rd Qu.	: 230.0	3rd Qu.	: 716.0				
Max.	:1415.0	Max.	:1727.0				

The `stat.desc` function in the `pastecs` package also gives us some additional descriptive stats. We can also specify a subset of our data; for example, suppose some of our data is numeric and some is not - we could specify just those stats that are numeric. In our dataset, even though all the values are numeric, some of the variables actually are not - `sex`, `smokstat`, and `vituse` use numeric codes to indicate categorical variables. So even though we technically could get descriptive stats on those, it wouldn't really make a lot of sense. Instead, we'll indicate column numbers to specify just those variables we want.

```
> stat.desc(dat[, 6:14])
```

	calories		fat		fiber		alcohol		cholesterol	
nbr.val	3.150000e+02	3.150000e+02	315.0000000	315.0000000	315.0000000	315.0000000	3.150000e+02			
nbr.null	0.000000e+00	0.000000e+00	0.0000000	0.0000000	111.0000000	0.000000e+00				
nbr.na	0.000000e+00	0.000000e+00	0.0000000	0.0000000	0.0000000	0.000000e+00				
min	4.452000e+02	1.440000e+01	3.1000000	0.0000000	0.0000000	3.770000e+01				
max	6.662200e+03	2.359000e+02	36.8000000	203.0000000	9.007000e+02					
range	6.217000e+03	2.215000e+02	33.7000000	203.0000000	8.630000e+02					
sum	5.659462e+05	2.426550e+04	4028.4000000	1033.0000000	7.637510e+04					
median	1.666800e+03	7.290000e+01	12.1000000	0.3000000	2.063000e+02					
mean	1.796655e+03	7.703333e+01	12.7885714	3.2793651	2.424606e+02					

SE.mean	3.833324e+01	1.906073e+00	0.3003223	0.6943156	7.436885e+00
CI.mean.0.95	7.542247e+01	3.750290e+00	0.5908985	1.3660991	1.463243e+01
var	4.628726e+05	1.144431e+03	28.4109518	151.8533627	1.742179e+04
std.dev	6.803474e+02	3.382944e+01	5.3301925	12.3228796	1.319916e+02
coef.var	3.786746e-01	4.391533e-01	0.4167934	3.7577029	5.443837e-01
	betadiet	retdiet	betaplasma	retplasma	
nbr.val	3.150000e+02	3.150000e+02	3.150000e+02	3.150000e+02	
nbr.null	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	
nbr.na	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	
min	2.140000e+02	3.000000e+01	0.000000e+00	1.790000e+02	
max	9.642000e+03	6.901000e+03	1.415000e+03	1.727000e+03	
range	9.428000e+03	6.871000e+03	1.415000e+03	1.548000e+03	
sum	6.884650e+05	2.623050e+05	5.981600e+04	1.898790e+05	
median	1.802000e+03	7.070000e+02	1.400000e+02	5.660000e+02	
mean	2.185603e+03	8.327143e+02	1.898921e+02	6.027905e+02	
SE.mean	8.304410e+01	3.320268e+01	1.031093e+01	1.176993e+01	
CI.mean.0.95	1.633932e+02	6.532785e+01	2.028724e+01	2.315789e+01	
var	2.172342e+06	3.472616e+05	3.348929e+04	4.363732e+04	
std.dev	1.473887e+03	5.892890e+02	1.830008e+02	2.088955e+02	
coef.var	6.743615e-01	7.076725e-01	9.637096e-01	3.465474e-01	

The psych package offers some additional options for exploring distributions. The describe() function provides overall summary statistics.

```
> describe(dat)
```

	vars	n	mean	sd	median	trimmed	mad	min	max
age	1	315	50.15	14.58	48.00	49.70	16.31	19.00	83.0
sex	2	315	1.87	0.34	2.00	1.96	0.00	1.00	2.0
smokstat	3	315	1.64	0.71	2.00	1.55	1.48	1.00	3.0
quetelet	4	315	26.16	6.01	24.74	25.31	4.83	16.33	50.4
vituse	5	315	1.97	0.86	2.00	1.96	1.48	1.00	3.0
calories	6	315	1796.65	680.35	1666.80	1733.51	564.87	445.20	6662.2
fat	7	315	77.03	33.83	72.90	74.07	31.13	14.40	235.9
fiber	8	315	12.79	5.33	12.10	12.32	4.60	3.10	36.8
alcohol	9	315	3.28	12.32	0.30	1.50	0.44	0.00	203.0
cholesterol	10	315	242.46	131.99	206.30	227.34	98.44	37.70	900.7
betadiet	11	315	2185.60	1473.89	1802.00	1972.76	1141.60	214.00	9642.0
retdiet	12	315	832.71	589.29	707.00	761.93	379.55	30.00	6901.0
betaplasma	13	315	189.89	183.00	140.00	157.95	87.47	0.00	1415.0
retplasma	14	315	602.79	208.90	566.00	586.42	188.29	179.00	1727.0
	range	skew	kurtosis	se					
age	64.00	0.30	-0.89	0.82					
sex	1.00	-2.15	2.62	0.02					
smokstat	2.00	0.65	-0.81	0.04					
quetelet	34.07	1.36	1.93	0.34					
vituse	2.00	0.07	-1.65	0.05					
calories	6217.00	1.73	7.91	38.33					
fat	221.50	1.09	1.93	1.91					
fiber	33.70	1.14	2.39	0.30					
alcohol	203.00	13.69	216.42	0.69					
cholesterol	863.00	1.47	3.30	7.44					
betadiet	9428.00	1.60	3.36	83.04					
retdiet	6871.00	4.43	37.19	33.20					
betaplasma	1415.00	3.53	16.79	10.31					
retplasma	1548.00	1.30	3.89	11.77					

It can also be useful to see how these stats differ among different groups. The `describeBy()` function allows us to see summary statistics broken down by a particular categorical variable. For example, we can see how our stats look for the men (coded as 1) and the women (coded as 2) or smoking status (1 for nonsmokers, 2 for current smokers, 3 for former smokers)

```
> describeBy(dat, sex)
```

group: 1

	vars	n	mean	sd	median	trimmed	mad	min	max
age	1	42	60.55	13.47	65.00	61.18	13.34	33.00	83.00
sex	2	42	1.00	0.00	1.00	1.00	0.00	1.00	1.00
smokstat	3	42	1.86	0.68	2.00	1.82	0.00	1.00	3.00
quetelet	4	42	26.27	4.18	25.19	25.85	3.10	19.41	41.65
vituse	5	42	2.26	0.91	3.00	2.32	0.00	1.00	3.00
calories	6	42	2155.79	916.57	2023.60	2074.01	622.62	827.90	6662.20
fat	7	42	93.89	33.65	94.45	94.09	28.39	32.80	166.00
fiber	8	42	13.41	4.84	12.10	13.06	4.60	4.70	26.30
alcohol	9	42	10.42	31.47	1.45	4.61	2.15	0.00	203.00
cholesterol	10	42	328.12	145.43	314.60	319.17	136.77	77.50	747.50
betadiet	11	42	2200.02	1072.13	2059.00	2158.62	1298.02	494.00	4387.00
retdiet	12	42	944.14	642.27	739.00	854.76	369.91	242.00	4041.00
betaplasma	13	42	148.67	133.60	104.00	128.12	70.42	21.00	751.00
retplasma	14	42	700.74	307.81	672.00	674.74	267.61	194.00	1727.00
	range		skew	kurtosis	se				
age	50.00		-0.43	-0.90	2.08				
sex	0.00		NaN	NaN	0.00				
smokstat	2.00		0.17	-0.93	0.11				
quetelet	22.24		1.37	2.56	0.64				
vituse	2.00		-0.52	-1.62	0.14				
calories	5834.30		2.71	11.33	141.43				
fat	133.20		0.02	-0.55	5.19				
fiber	21.60		0.64	-0.31	0.75				
alcohol	203.00		5.45	30.41	4.86				
cholesterol	670.00		0.60	0.10	22.44				
betadiet	3893.00		0.30	-1.09	165.43				
retdiet	3799.00		2.78	10.49	99.10				
betaplasma	730.00		2.39	7.52	20.61				
retplasma	1533.00		0.95	1.25	47.50				

-----

group: 2

	vars	n	mean	sd	median	trimmed	mad	min	max
age	1	273	48.55	14.09	46.00	47.89	13.34	19.00	83.0
sex	2	273	2.00	0.00	2.00	2.00	0.00	2.00	2.0
smokstat	3	273	1.60	0.71	1.00	1.51	0.00	1.00	3.0
quetelet	4	273	26.14	6.25	24.26	25.24	4.83	16.33	50.4
vituse	5	273	1.92	0.85	2.00	1.90	1.48	1.00	3.0
calories	6	273	1741.40	620.27	1628.50	1682.11	559.38	445.20	4373.6
fat	7	273	74.44	33.16	68.40	71.13	26.69	14.40	235.9
fiber	8	273	12.69	5.40	12.10	12.19	4.60	3.10	36.8
alcohol	9	273	2.18	4.12	0.20	1.22	0.30	0.00	35.0
cholesterol	10	273	229.28	124.97	195.60	214.10	92.07	37.70	900.7
betadiet	11	273	2183.38	1527.90	1734.00	1946.19	1091.19	214.00	9642.0
retdiet	12	273	815.57	580.08	706.00	747.02	379.55	30.00	6901.0
betaplasma	13	273	196.23	188.86	144.00	162.95	85.99	0.00	1415.0
retplasma	14	273	587.72	185.43	561.00	576.54	171.98	179.00	1517.0

	range	skew	kurtosis	se
age	64.00	0.41	-0.70	0.85
sex	0.00	NaN	NaN	0.00
smokstat	2.00	0.73	-0.73	0.04
quetelet	34.07	1.34	1.70	0.38
vituse	2.00	0.15	-1.59	0.05
calories	3928.40	0.96	1.21	37.54
fat	221.50	1.32	2.88	2.01
fiber	33.70	1.21	2.67	0.33
alcohol	35.00	3.37	16.62	0.25
cholesterol	863.00	1.70	4.81	7.56
betadiet	9428.00	1.64	3.29	92.47
retdiet	6871.00	4.76	43.34	35.11
betaplasma	1415.00	3.53	16.34	11.43
retplasma	1338.00	1.04	3.03	11.22

```
> describeBy(dat, smokstat)
```

group: 1

	vars	n	mean	sd	median	trimmed	mad	min	max
age	1	157	51.22	15.02	49.00	50.98	17.79	19.00	83.0
sex	2	157	1.92	0.28	2.00	2.00	0.00	1.00	2.0
smokstat	3	157	1.00	0.00	1.00	1.00	0.00	1.00	1.0
quetelet	4	157	26.73	6.86	24.26	25.77	5.18	16.64	50.4
vituse	5	157	1.84	0.85	2.00	1.80	1.48	1.00	3.0
calories	6	157	1713.39	592.18	1631.00	1664.19	564.87	445.20	4373.6
fat	7	157	72.19	31.99	65.70	69.05	24.31	14.40	235.9
fiber	8	157	13.16	5.70	12.70	12.57	4.30	3.70	36.8
alcohol	9	157	1.67	3.36	0.10	0.88	0.15	0.00	21.0
cholesterol	10	157	228.39	134.23	195.80	209.80	91.77	37.70	900.7
betadiet	11	157	2193.39	1386.82	1976.00	2022.09	1220.18	330.00	8046.0
retdiet	12	157	838.85	659.13	707.00	759.96	379.55	30.00	6901.0
betaplasma	13	157	206.05	193.21	158.00	174.82	105.26	0.00	1415.0
retplasma	14	157	583.31	187.64	564.00	573.84	177.91	187.00	1517.0

	range	skew	kurtosis	se
age	64.00	0.20	-0.92	1.20
sex	1.00	-3.00	7.04	0.02
smokstat	0.00	NaN	NaN	0.00
quetelet	33.77	1.25	1.13	0.55
vituse	2.00	0.31	-1.56	0.07
calories	3928.40	1.12	2.59	47.26
fat	221.50	1.53	4.41	2.55
fiber	33.10	1.43	3.41	0.45
alcohol	21.00	3.10	11.38	0.27
cholesterol	863.00	1.94	5.63	10.71
betadiet	7716.00	1.29	1.92	110.68
retdiet	6871.00	5.36	44.05	52.60
betaplasma	1415.00	3.71	18.48	15.42
retplasma	1330.00	0.98	3.15	14.98

group: 2

	vars	n	mean	sd	median	trimmed	mad	min	max
age	1	115	50.77	13.97	48.00	50.34	16.31	22.00	83.00
sex	2	115	1.81	0.40	2.00	1.88	0.00	1.00	2.00
smokstat	3	115	2.00	0.00	2.00	2.00	0.00	2.00	2.00

quetelet	4	115	25.93	4.99	24.94	25.35	3.79	18.58	44.21
vituse	5	115	2.03	0.86	2.00	2.03	1.48	1.00	3.00
calories	6	115	1861.18	643.53	1751.10	1823.99	670.58	659.30	3711.00
fat	7	115	80.93	35.44	76.50	78.02	32.77	20.40	202.70
fiber	8	115	13.10	4.91	12.50	12.83	5.19	3.10	26.50
alcohol	9	115	3.85	5.82	1.00	2.62	1.48	0.00	35.00
cholesterol	10	115	250.42	121.69	216.70	241.92	108.53	46.30	747.50
betadiet	11	115	2355.23	1625.25	1945.00	2101.06	1168.29	241.00	9642.00
retdiet	12	115	838.76	529.63	728.00	772.25	366.20	125.00	4041.00
betaplasma	13	115	193.47	191.64	135.00	156.77	78.58	16.00	1212.00
retplasma	14	115	644.24	231.17	587.00	622.61	207.56	216.00	1727.00

	range	skew	kurtosis	se
age	61.00	0.28	-0.98	1.30
sex	1.00	-1.55	0.40	0.04
smokstat	0.00	NaN	NaN	0.00
quetelet	25.63	1.29	1.93	0.47
vituse	2.00	-0.05	-1.67	0.08
calories	3051.70	0.55	-0.32	60.01
fat	182.30	0.88	0.94	3.30
fiber	23.40	0.54	-0.30	0.46
alcohol	35.00	2.33	6.93	0.54
cholesterol	701.20	0.87	1.02	11.35
betadiet	9401.00	1.71	3.60	151.56
retdiet	3916.00	2.40	10.32	49.39
betaplasma	1196.00	2.87	9.69	17.87
retplasma	1511.00	1.48	4.04	21.56

-----

group: 3

	vars	n	mean	sd	median	trimmed	mad	min	max
age	1	43	44.53	13.51	42.00	43.43	11.86	25.00	74.00
sex	2	43	1.84	0.37	2.00	1.91	0.00	1.00	2.00
smokstat	3	43	3.00	0.00	3.00	3.00	0.00	3.00	3.00
quetelet	4	43	24.69	4.92	23.38	24.20	4.20	16.33	41.65
vituse	5	43	2.26	0.82	2.00	2.31	1.48	1.00	3.00
calories	6	43	1928.10	989.40	1662.70	1800.09	625.21	784.40	6662.20
fat	7	43	84.31	34.08	76.60	83.32	34.84	25.20	164.30
fiber	8	43	10.60	4.54	9.80	10.12	4.60	4.70	23.30
alcohol	9	43	7.64	31.17	0.20	1.75	0.30	0.00	203.00
cholesterol	10	43	272.53	145.92	239.20	255.75	129.73	78.30	718.80
betadiet	11	43	1703.53	1269.14	1301.00	1518.74	722.03	214.00	7026.00
retdiet	12	43	794.14	468.38	677.00	759.89	428.47	141.00	2118.00
betaplasma	13	43	121.33	78.81	105.00	110.80	51.89	25.00	418.00
retplasma	14	43	563.07	206.58	539.00	540.34	179.39	179.00	1193.00

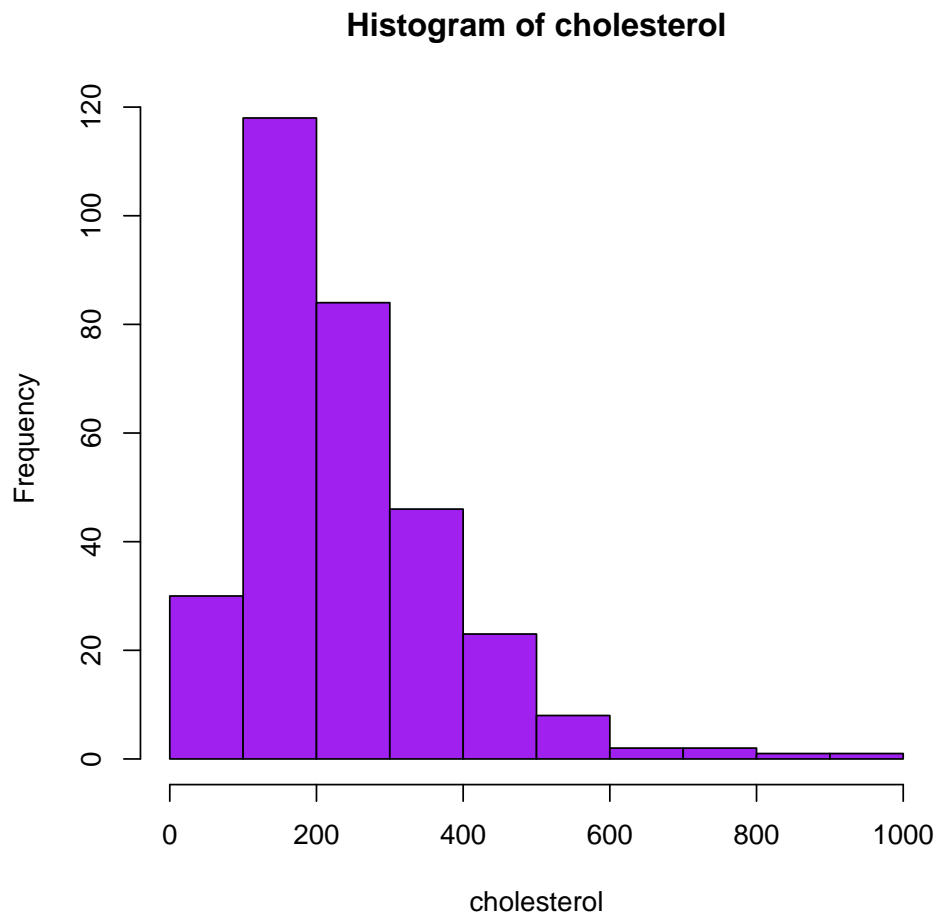
	range	skew	kurtosis	se
age	49.00	0.73	-0.36	2.06
sex	1.00	-1.76	1.14	0.06
smokstat	0.00	NaN	NaN	0.00
quetelet	25.32	1.11	1.64	0.75
vituse	2.00	-0.48	-1.38	0.12
calories	5877.80	2.57	9.60	150.88
fat	139.10	0.31	-0.98	5.20
fiber	18.60	0.90	0.19	0.69
alcohol	203.00	5.73	32.91	4.75
cholesterol	640.50	1.06	0.65	22.25

betadiet	6812.00	1.95	4.99	193.54
retdiet	1977.00	0.70	-0.34	71.43
betaplasma	393.00	1.66	3.50	12.02
retplasma	1014.00	1.04	1.16	31.50

## 2.1 Exploratory Data Visualization

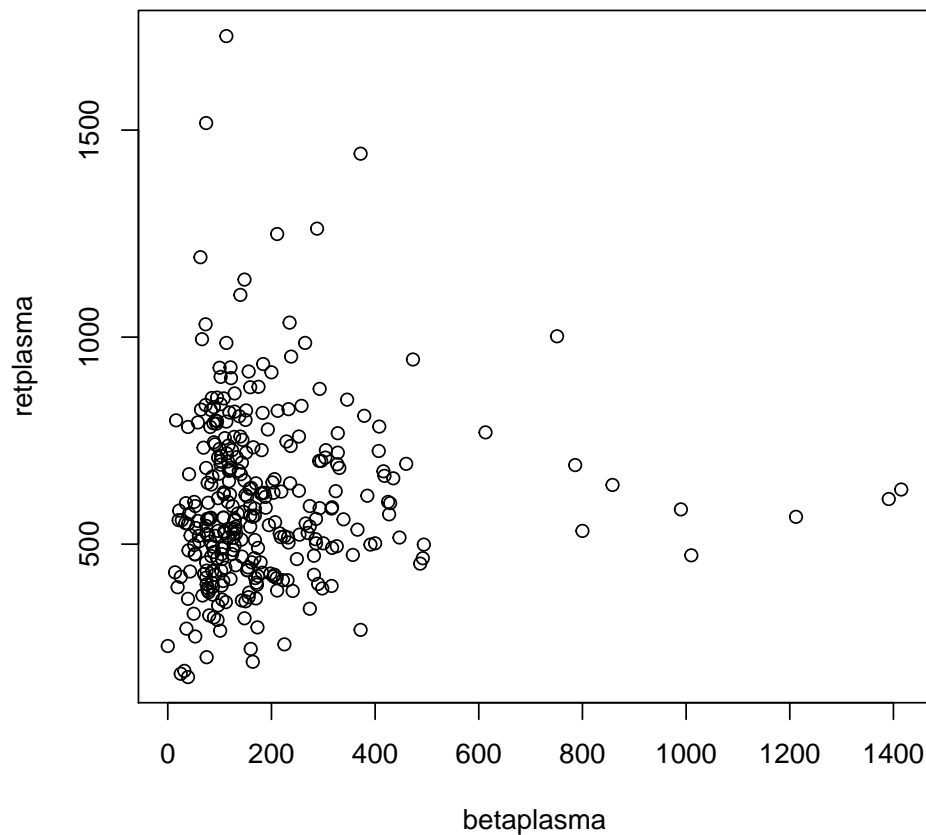
It might also be useful to look at some plots to give us a sense of the distribution of our data. R's basic plotting functionality can make several different types of plots for us. Let's start with a basic histogram of cholesterol levels. Just for fun, let's make it purple!

```
> hist(cholesterol, col = "purple")
```



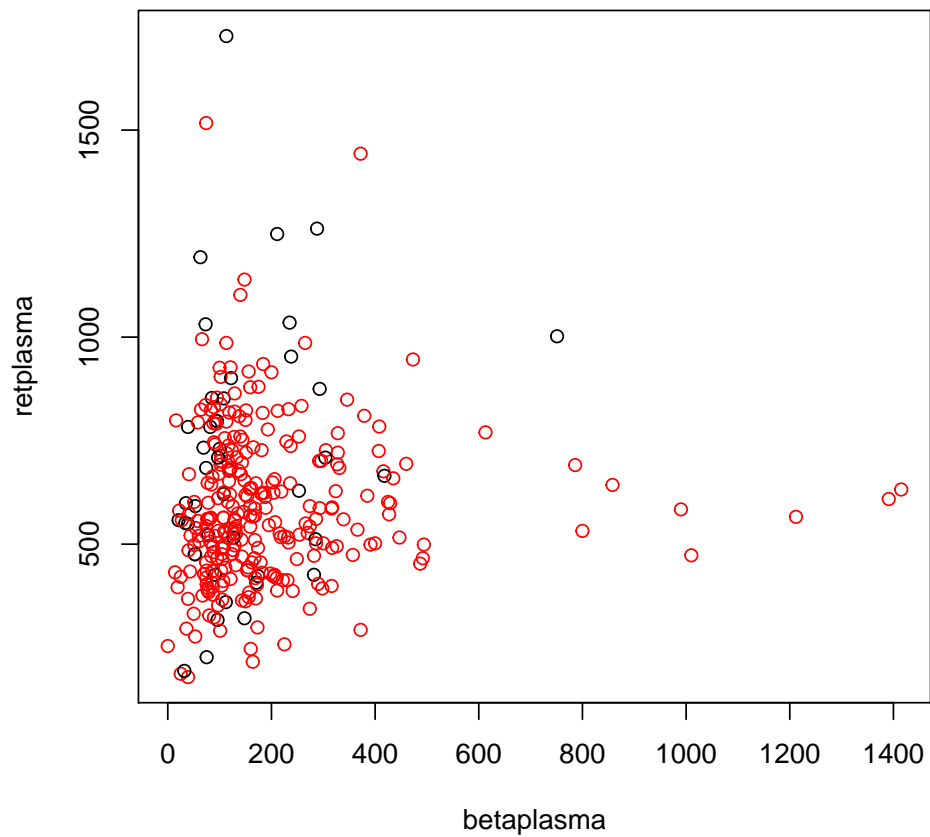
Next let's do a scatterplot to look at distributions of two variables.

```
> plot(betaplasma, retplasma)
```



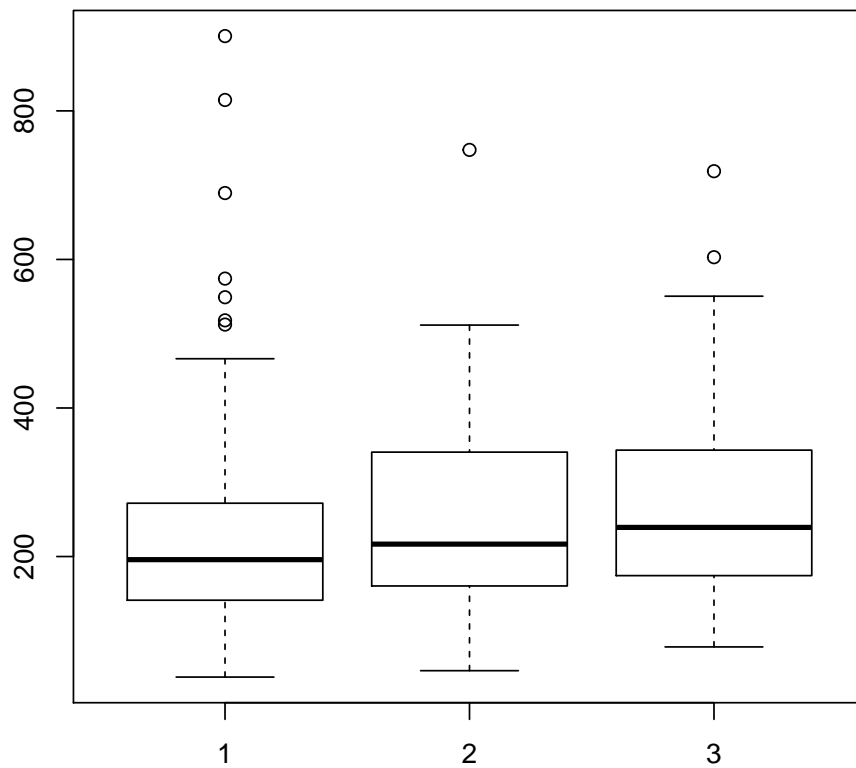
I can also color my points depending on which group they're in. Let's have our men and women shown as different colors. Note that I've specified I want to treat sex as a factor value, instead of a numeric value, which it is now. This is because this variable is truly a categorical variable even though it looks like a number to R. Also note that we don't have a legend showing us which color corresponds to which value of our variable. It is possible (though somewhat complex) to add a legend. It's much easier to create this kind of a plot using the ggplot2 package, which will be covered in a separate class (see <http://nihlibrary.campusguides.com/dataservices/ggplot>)

```
> plot(betaplasma, retplasma, col = factor(sex))
```



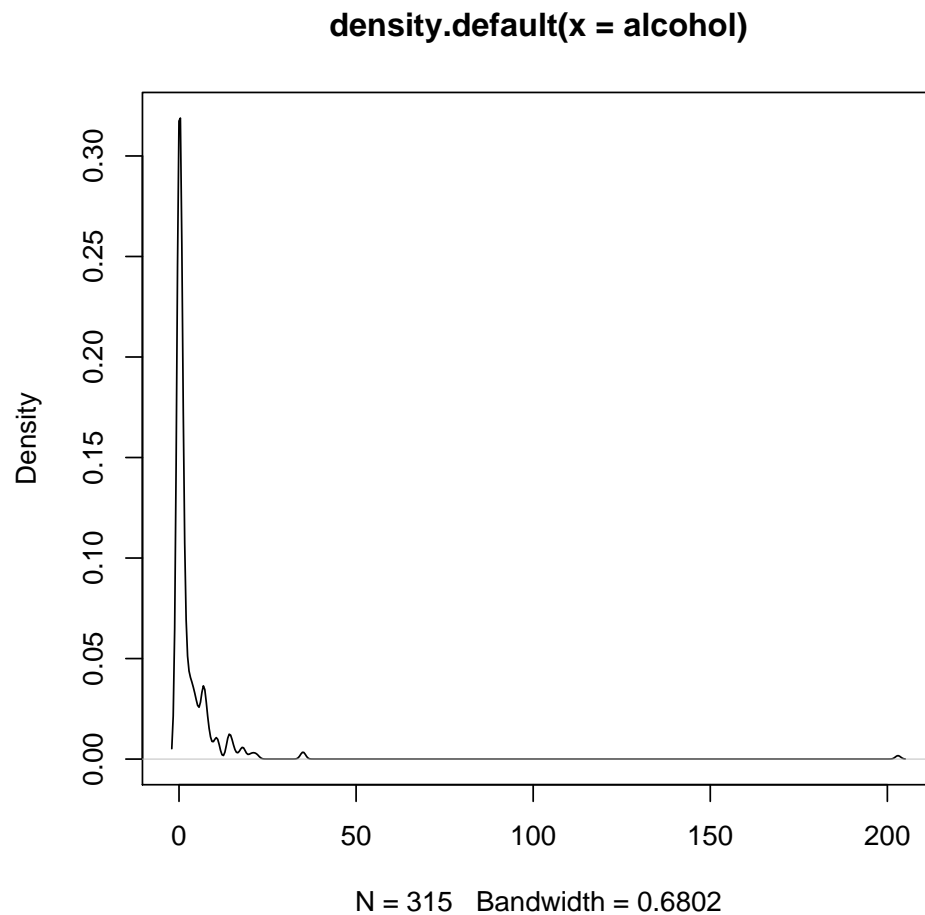
We can also do a boxplot - let's make one looking at distribution of cholesterol by smoking status. I'm doing the same thing here with the factor versus numeric variable for smokstat.

```
> boxplot(cholesterol ~ factor(smokstat))
```



Finally, let's do a density plot, in this case, for alcohol consumption by week. We do so first by telling R to calculate the density of alcohol, then plot it, which we can do all in one line.

```
> plot(density(alcohol))
```



This density plot of alcohol consumptions should raise a red flag - apparently there is someone who has reported drinking 203 drinks per week, which would work out to 29 drinks a day. I'm not at all confident that this is true; I think it's way more likely that someone made a data entry mistake than that someone is drinking 203 drinks a week. Let's find out which row has that observation and remove it from our dataset.

```
> which(alcohol == 203)
```

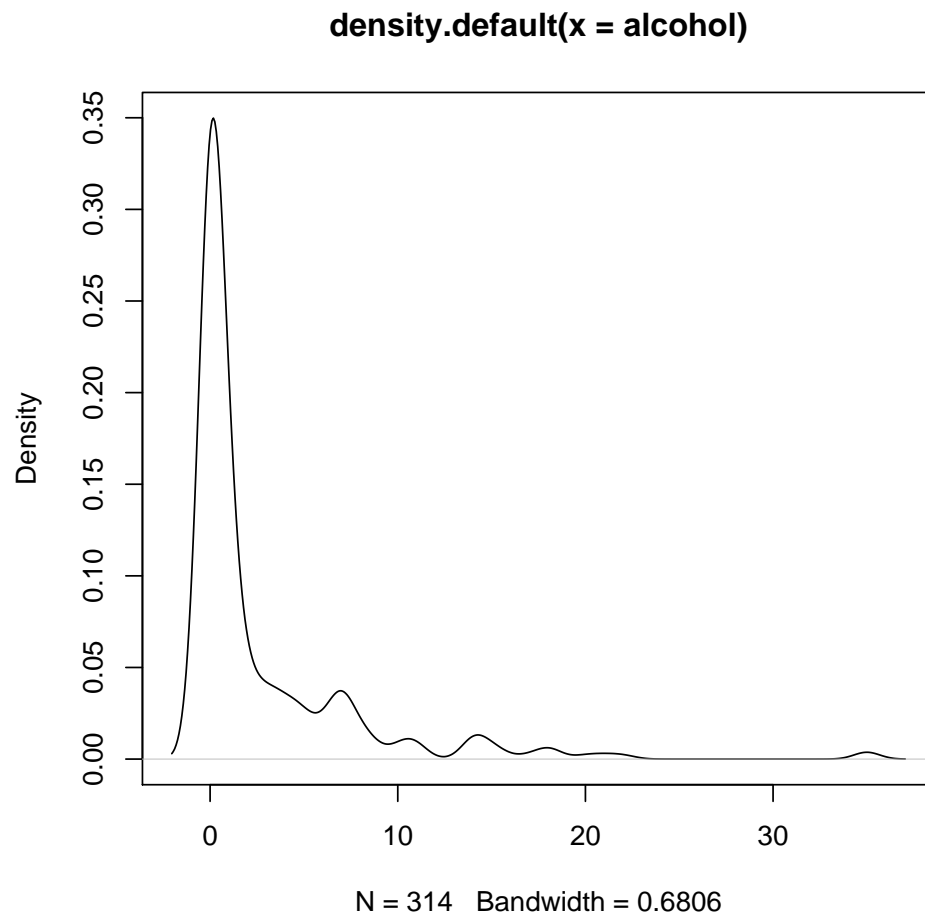
```
[1] 62
```

Here I've asked R to tell me which row contains an observation of alcohol consumption equal to 203, and it has determined that it's row 62. We'll detach the data frame, make the change, and then reattach it.

```
> detach(dat)
> dat <- dat[-62, ]
> attach(dat)
```

You'll notice now that you have one fewer observation than before in your data frame. Let's also redraw our density plot to see if this looks more reasonable now.

```
> plot(density(alcohol))
```



Much better!

## 2.2 Am I Normal?

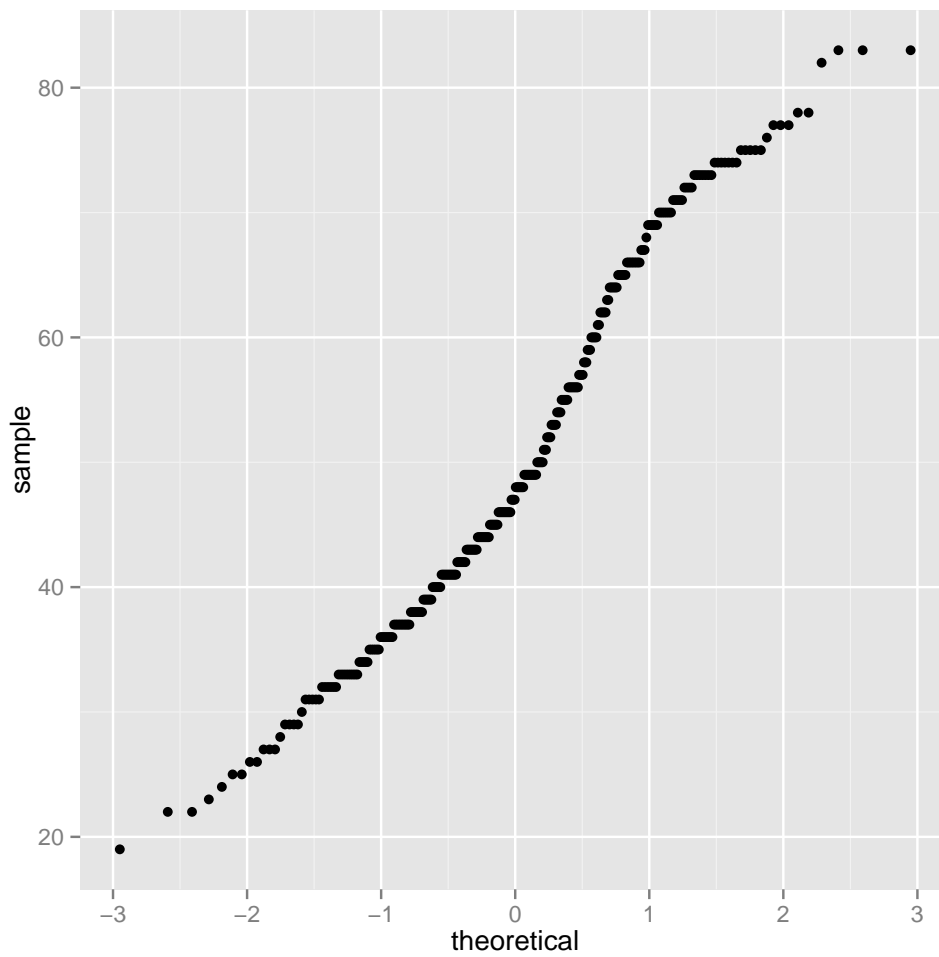
Many of the statistical tests we'll do assume "normality" of our data - that is, it's assumed our data are normally distributed. In practice, this is not always the case. We should test our data to see if they are normally distributed before we select which statistical methods we'll use.

First, we can visually check to see if our groups are normally distributed. One way is to look at a Q-Q plot, which looks at cumulative values of our data versus cumulative values of a normal distribution. If the data are normally distributed, we should see a diagonal line, because our actual scores will correspond to our expected (ie, normal) scores. We're going to use the `qplot` function from `ggplot2`, which is a little bit easier than using base R graphics, like we've been doing. Let's look at alcohol consumption.

```
> qplot(sample = alcohol, stat = "qq")
```

Nope! Not even close.

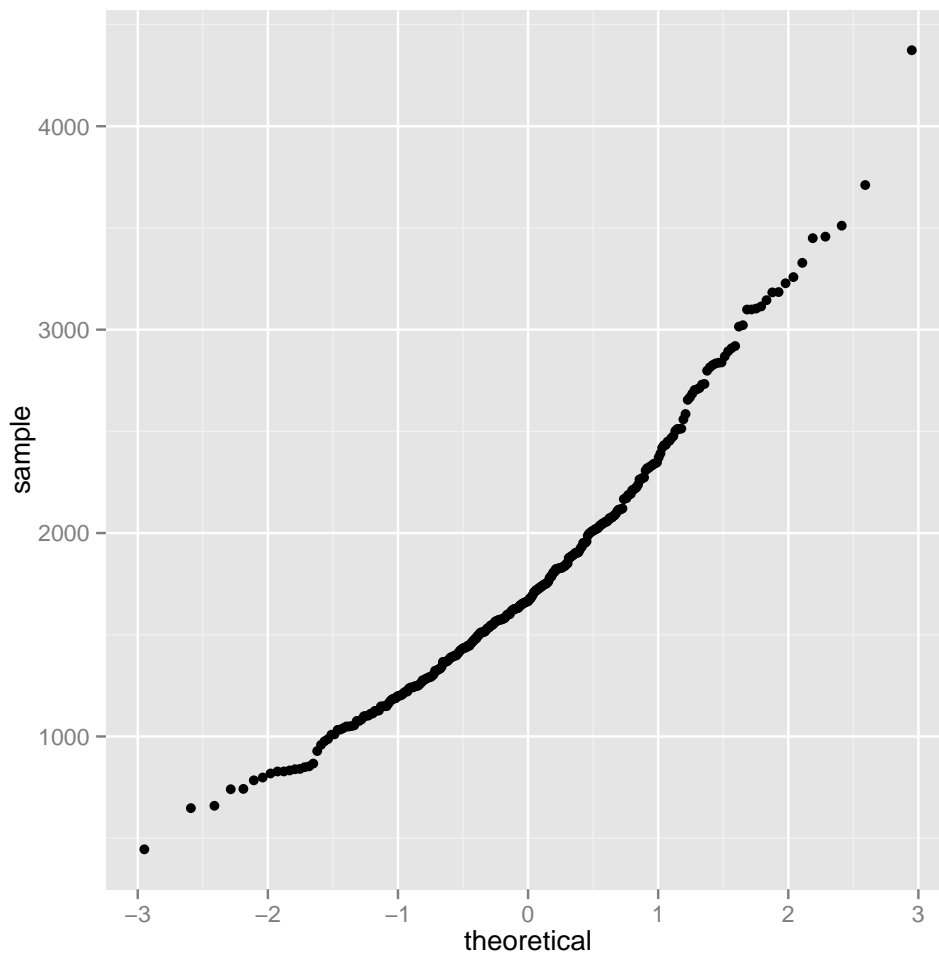
```
> qplot(sample = age, stat = "qq")
```



to normal distribution.

This looks a little closer

```
> qqplot(sample = calories, stat = "qq")
```



Not bad - pretty close!

We can also quantify our distribution numerically. Let's look back at our describe function results again.

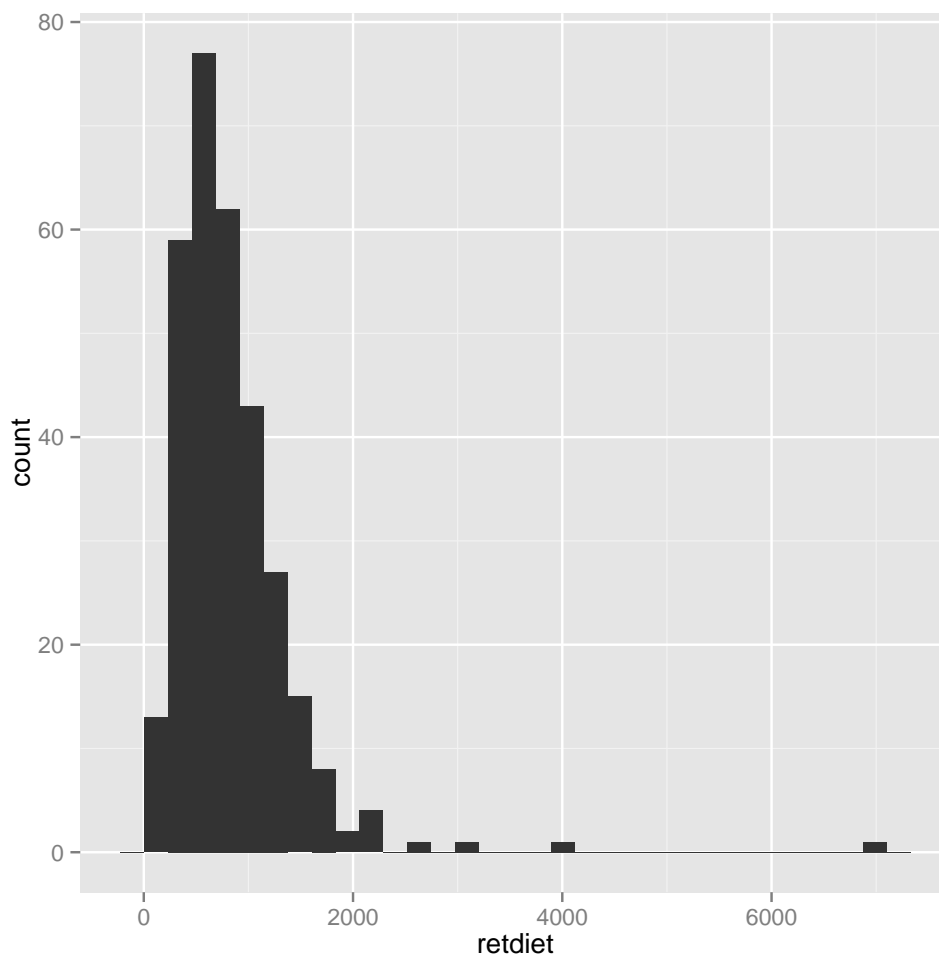
```
> describe(dat)
```

	vars	n	mean	sd	median	trimmed	mad	min	max
age	1	314	50.10	14.57	47.50	49.64	15.57	19.00	83.0
sex	2	314	1.87	0.34	2.00	1.96	0.00	1.00	2.0
smokstat	3	314	1.63	0.71	1.50	1.54	0.74	1.00	3.0
quetelet	4	314	26.17	6.02	24.74	25.32	4.87	16.33	50.4
vituse	5	314	1.96	0.86	2.00	1.95	1.48	1.00	3.0
calories	6	314	1781.16	623.28	1665.05	1729.65	564.94	445.20	4373.6
fat	7	314	76.76	33.52	72.90	73.88	31.06	14.40	235.9
fiber	8	314	12.79	5.34	12.10	12.32	4.67	3.10	36.8
alcohol	9	314	2.64	4.95	0.30	1.47	0.44	0.00	35.0
cholesterol	10	314	241.31	130.62	206.20	226.56	98.00	37.70	900.7
betadiet	11	314	2183.35	1475.70	1795.00	1969.11	1130.48	214.00	9642.0
retdiet	12	314	831.02	589.46	707.00	759.54	379.55	30.00	6901.0
betaplasma	13	314	190.19	183.22	140.00	158.19	88.21	0.00	1415.0
retplasma	14	314	603.70	208.60	566.00	587.21	187.55	179.00	1727.0
	range	skew	kurtosis	se					
age	64.00	0.30	-0.88	0.82					
sex	1.00	-2.18	2.77	0.02					
smokstat	2.00	0.65	-0.80	0.04					

quetelet	34.07	1.36	1.91	0.34
vituse	2.00	0.07	-1.65	0.05
calories	3928.40	0.82	0.79	35.17
fat	221.50	1.10	2.02	1.89
fiber	33.70	1.14	2.37	0.30
alcohol	35.00	3.12	12.80	0.28
cholesterol	863.00	1.48	3.46	7.37
betadiet	9428.00	1.60	3.36	83.28
retdiet	6871.00	4.45	37.32	33.27
betaplasma	1415.00	3.52	16.74	10.34
retplasma	1548.00	1.31	3.92	11.77

Skew and kurtosis are what we want to look at here. Skew is a measure of symmetry or lack of. Normal distribution has skew = 0; negative skew means the data are skewed left, while positive skew means the data are skewed right. We can see an example of this by looking at sex (negative skew, long tail to the left) and retdiet (positive skew, long tail to the right) histograms.

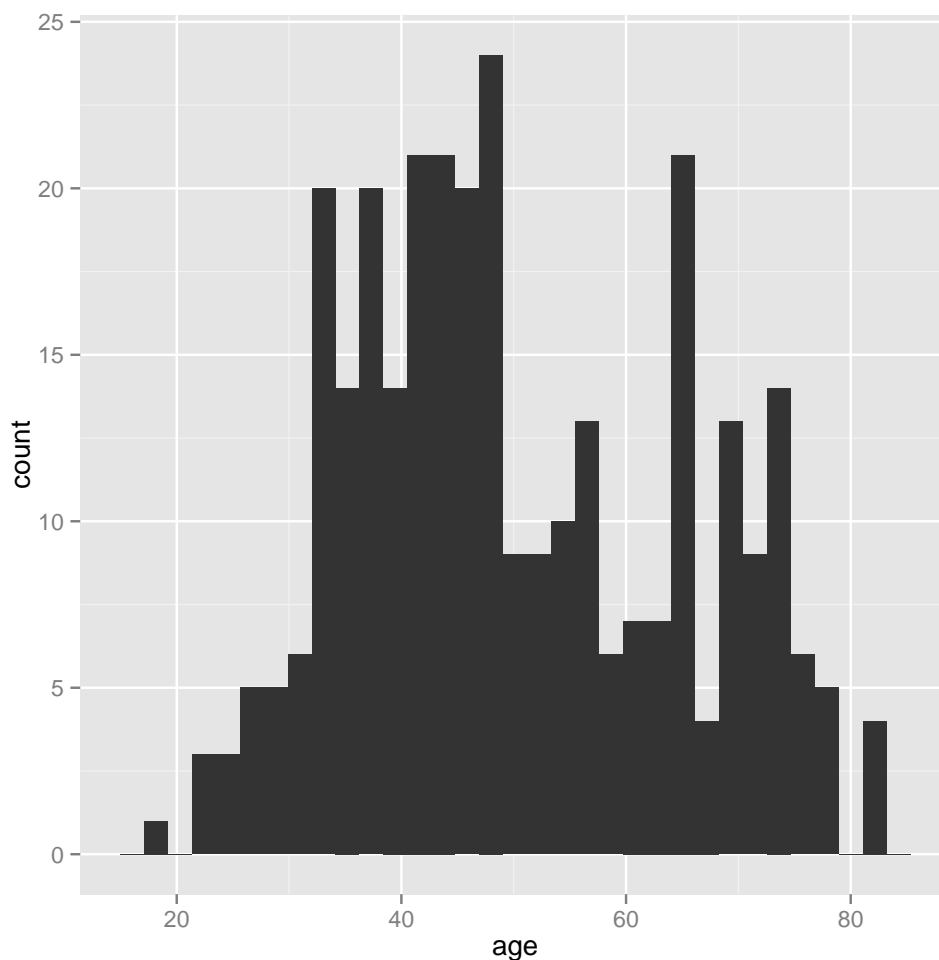
```
> qplot(retdiet, geom = "histogram")
> qplot(sex, geom = "histogram")
```



Kurtosis is a measure of whether the data are peaked or flat compared to normal distribution. A high kurtosis value means there is a peak near the mean and a rapid decline, so there's a heavy tail. Low kurtosis means that there's not a steep decline from the mean. For normal distribution, kurtosis is 3.

Age gives an example of low kurtosis - the data are pretty flat all the way across. On the other hand, retidiet gives an example of high kurtosis - there's a very high peak and then the data rapidly decline.

```
> qplot(age, geom = "histogram")
> qplot(retidiet, geom = "histogram")
```



I might also want to check this for two different groups, like my men/women and my smokers/non-smokers. I can do that using the `by()` function, specifying the data I want to use (including one or more variables), the variable I want to split on, and the stats I want (in this case, `describe`).

```
> by(retidiet, sex, describe)
```

```
sex: 1
  vars  n  mean    sd median trimmed   mad min  max range skew kurtosis
1    1 41 933.9 646.77   735  839.33 363.24 242 4041  3799 2.83    10.61
  se
1 101.01
-----
sex: 2
  vars  n  mean    sd median trimmed   mad min  max range skew kurtosis
1    1 273 815.57 580.08   706  747.02 379.55  30 6901  6871 4.76    43.34
  se
1 35.11
```

```
> by(cbind(retdiet, retplasma, fiber), sex, describe)

INDICES: 1
      vars  n   mean      sd median trimmed   mad   min    max  range skew
retdiet    1 41 933.90 646.77  735.0  839.33 363.24 242.0 4041.0 3799.0 2.83
retplasma  2 41 710.10 305.52  679.0  684.27 256.49 194.0 1727.0 1533.0 0.95
fiber      3 41  13.47   4.89   12.9   13.11   5.34   4.7   26.3   21.6 0.60
      kurtosis      se
retdiet    10.61 101.01
retplasma   1.30  47.71
fiber      -0.38   0.76
-----

INDICES: 2
      vars  n   mean      sd median trimmed   mad   min    max  range skew
retdiet    1 273 815.57 580.08  706.0  747.02 379.55  30.0 6901.0 6871.0 4.76
retplasma  2 273 587.72 185.43  561.0  576.54 171.98 179.0 1517.0 1338.0 1.04
fiber      3 273  12.69   5.40   12.1   12.19   4.60   3.1   36.8   33.7 1.21
      kurtosis      se
retdiet    43.34 35.11
retplasma   3.03 11.22
fiber       2.67  0.33
```

Another way of looking at whether our data follow a normal distribution is to do a Shapiro Wilk test. In this test, we will see a large p-value (greater than .05) if our data are *not* statistically different from normal distribution, and a low p-value if our data *are* statistically different from normal.

```
> shapiro.test(age)

Shapiro-Wilk normality test
```

```
data:  age
W = 0.9657, p-value = 9.099e-07
```

We can test two different groups if we want as well. Interestingly, in this case fiber consumption in men is normally distributed, while in women it is not.

```
> by(fiber, sex, shapiro.test)

sex: 1

Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.9529, p-value = 0.08856

-----

sex: 2

Shapiro-Wilk normality test

data:  dd[x, ]
W = 0.9294, p-value = 4.187e-10
```

### 3 Comparing Two Groups

We often want to compare two groups, like two different populations (men and women, smokers and non-smokers, etc) or groups who had different interventions (treatment and control group).

Before we can do this, we need to change our data up a little bit to get it in format we can use. There are lots of different ways you could do this, but I'm going to do it by splitting my male and female participants up into two different data frames that I can compare.

```
> female <- subset(dat, sex == 2)
> male <- subset(dat, sex == 1)
```

As a side note, take a look at the number of observations in our two groups: 273 women and 41 men. If I were doing this for real, I'd want to have my groups more balanced, but since I'm just doing this for demonstration purposes to teach you how to use this, I'm not going to worry about it. Also, it's very important to note that we would not do parametric tests, like some of the ones we're about to do, on data that are NOT normally distributed.

One test to compare two groups is to compare two variances. We can do this using the `var.test()` function. For example, let's look at variance in alcohol consumption between the two groups. Check the p-values to see if the results are statistically significant.

```
> var.test(female$alcohol, male$alcohol)
```

F test to compare two variances

```
data: female$alcohol and male$alcohol
F = 0.2599, num df = 272, denom df = 40, p-value = 3.706e-11
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1550058 0.4006387
sample estimates:
ratio of variances
      0.259915
```

```
> var.test(female$age, male$age)
```

F test to compare two variances

```
data: female$age and male$age
F = 1.071, num df = 272, denom df = 40, p-value = 0.8229
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.6387035 1.6508367
sample estimates:
ratio of variances
      1.070983
```

By default, `var.test` uses a 95 per cent confidence interval, but we could also set our own if we want it to be higher or lower.

```
> var.test(female$betaplasma, male$betaplasma, conf.level = 0.98)
```

F test to compare two variances

```
data: female$betaplasma and male$betaplasma
F = 1.9571, num df = 272, denom df = 40, p-value = 0.01192
alternative hypothesis: true ratio of variances is not equal to 1
98 percent confidence interval:
 1.054573 3.263358
sample estimates:
ratio of variances
      1.957138
```

Another way to compare two groups is to compare two means. One way we can do this is by using the Student's t-test. The small p-value in our first example here tells us that the mean calories consumed by women is significantly different from the mean for men. Although the variance between male/female age was not significantly different, the difference in means is - the male sample is much older than the female sample.

```
> t.test(female$calories, male$calories)

Welch Two Sample t-test

data:  female$calories and male$calories
t = -3.0872, df = 54.473, p-value = 0.003176
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -502.1602 -106.7788
sample estimates:
mean of x mean of y
 1741.404  2045.873

> t.test(female$age, male$age)

Welch Two Sample t-test

data:  female$age and male$age
t = -5.1903, df = 53.698, p-value = 3.294e-06
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -16.487918 -7.298556
sample estimates:
mean of x mean of y
 48.54579  60.43902
```

Sometimes the Students t-test is not appropriate - check out a statistical text to find out more about when this might be the case. Another alternative is the Wilcoxon Rank-Sum Test. This is what's known as a non-parametric test and it does not assume normal distribution of the data.

```
> wilcox.test(female$cholesterol, male$cholesterol)

Wilcoxon rank sum test with continuity correction

data:  female$cholesterol and male$cholesterol
W = 3211, p-value = 1.083e-05
alternative hypothesis: true location shift is not equal to 0

> wilcox.test(female$fiber, male$fiber)

Wilcoxon rank sum test with continuity correction

data:  female$fiber and male$fiber
W = 5037, p-value = 0.3024
alternative hypothesis: true location shift is not equal to 0
```

When it makes sense to put your data in a contingency table, you can use a chi-square test to test independence. For example, we could test whether smoking and alcohol consumption are independent in this sample. To do any sort of contingency tests, we need to create a contingency table. We'll be looking at the whole sample, not the male/female subsets.

First we need to create dummy variables for smoking and drinking status. Then we'll create a table based on those variables.

```

> dat$smoke[dat$smokstat == 1] <- "non-smoker"
> dat$smoke[dat$smokstat != 1] <- "current or former smoker"
> dat$drink[dat$alcohol == 0] <- "non-drinker"
> dat$drink[dat$alcohol > 0] <- "drinker"
> counts <- table(dat$smoke, dat$drink)

```

Now that we have our table, we can do some tests on it. Our chi squared test has a high p-value, indicating no significant relationship between drinking and smoking. The Fisher's exact test is used when one of the samples has less than 5 observations. We get the same result as for the chi squared test.

```

> chisq.test(counts)

```

Pearson's Chi-squared test with Yates' continuity correction

```

data: counts
X-squared = 3.5674, df = 1, p-value = 0.05892

```

```

> fisher.test(counts)

```

Fisher's Exact Test for Count Data

```

data: counts
p-value = 0.05869
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.9837894 2.6417999
sample estimates:
odds ratio
 1.608117

```

sectionRegression Regression analysis allows us to predict one variable based on one or more other variables. Simple regression predicts an outcome variable from one predictor variable; multiple regression predicts an outcome variable from several predictor variables.

To start, we'll do a simple regression to see if the amount of dietary beta-carotene predicts plasma beta-carotene. With the `lm` function, the outcome variable comes first, then the predictor variable(s). Then we use the `summary` or `summary,aov` functions to get the results of our analyses.

```

> fit <- lm(betaplasma ~ betadiet)
> summary(fit)

```

Call:

```
lm(formula = betaplasma ~ betadiet)
```

Residuals:

Min	1Q	Median	3Q	Max
-286.92	-93.03	-37.98	33.35	1155.92

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.290e+02	1.804e+01	7.152	6.11e-12 ***
betadiet	2.803e-02	6.847e-03	4.093	5.43e-05 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 178.8 on 312 degrees of freedom

Multiple R-squared: 0.05096, Adjusted R-squared: 0.04792

F-statistic: 16.75 on 1 and 312 DF, p-value: 5.426e-05

```
> summary.aov(fit)

              Df Sum Sq Mean Sq F value    Pr(>F)
betadiet         1  535431   535431   16.75 5.43e-05 ***
Residuals       312 9971363    31959
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

What does this tell us? First of all, our Rsquared result of 0.05096 tells us that dietary beta carotene accounts for just 5 per cent of the variation in plasma beta carotene. We can also compare our F-value to the critical value of F, which we would have to look up in an F-distribution table. For 1 and 312, our F value is larger than the critical value, and our very small p-value tells us it's unlikely this happened by chance. Therefore, we can reject the null hypothesis that dietary beta carotene has no effect on plasma beta carotene (even though that effect is pretty small).

Using the `lm.beta()` function from the `QuantPsyc` package, we can get the standardized beta value, which tells us how much the outcome variable will change as the result of one standard deviation change in the predictor variable. Using the standard deviation of `betadiet`, we can calculate the actual value of plasma beta carotene.

```
> lm.beta(fit)

betadiet
0.2257442

> sd(betadiet)

[1] 1475.696

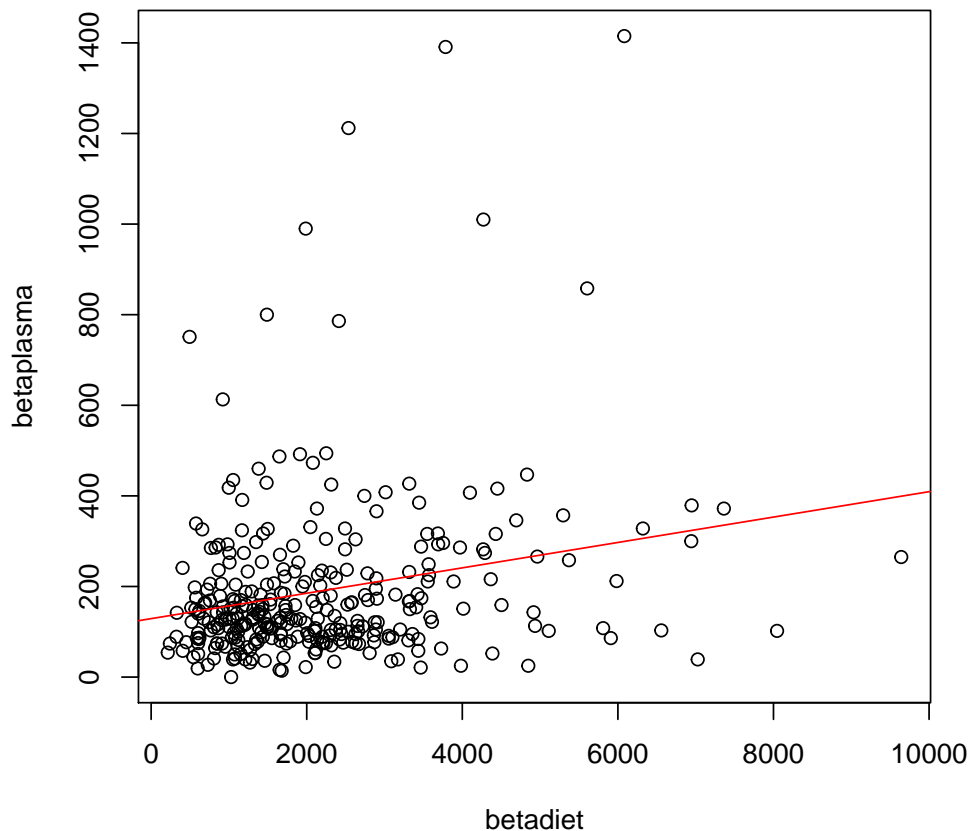
> sd(betaplasma) * lm.beta(fit)

betadiet
41.35991
```

This tells us that, for every increase by one standard deviation, or 1475.7 mcg per day of dietary beta carotene, we can expect beta plasma carotene to increase by .226 standard deviations. Since the standard deviation of plasma beta carotene is 183.2, we can expect our increase to be 41.34 ng/ml.

We can also create a simple plot to show us how our model looks and create some plots to check our model. First, we can create a scatterplot of our data and add a regression line based on the model we just created. There is also a set of four standard plots we can create from our model. If we want to see them all together, we can modify our graphics to show a 2x2 grid.

```
> plot(betaplasma ~ betadiet)
> abline(fit, col = "red")
> par(mfrow = c(2, 2))
> plot(fit)
> par(mfrow = c(1, 1))
```



There are some good descriptions of how to interpret these plots available online. Check out <http://stats.stackexchange.com/questions/58141/interpreting-plot-lm> and <http://www.r-bloggers.com/model-validation-interpreting-residual-plots/>

Since the effect of dietary beta carotene is pretty small, let's see if there are other predictor variables that have an effect by doing a multiple regression. We'll look at how dietary beta carotene, dietary retinol, vitamin use, and the amounts of fat and fiber consumed effect plasma beta carotene

```
> fit2 <- lm(betaplasma ~ betadiet + fat + fiber + retdiet)
> summary(fit2)
```

Call:

```
lm(formula = betaplasma ~ betadiet + fat + fiber + retdiet)
```

Residuals:

Min	1Q	Median	3Q	Max
-260.95	-87.50	-40.90	28.72	1125.93

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )	
(Intercept)	130.002854	30.987273	4.195	3.56e-05	***
betadiet	0.017941	0.007676	2.337	0.020055	*
fat	-0.850068	0.331581	-2.564	0.010830	*
fiber	7.456160	2.203953	3.383	0.000809	***

```
retdiet      -0.010982   0.018563  -0.592 0.554566
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 175 on 309 degrees of freedom
Multiple R-squared:  0.09914,    Adjusted R-squared:  0.08748
F-statistic: 8.501 on 4 and 309 DF,  p-value: 1.612e-06
```

What can we tell from this? When we looked at just dietary beta carotene, our R squared value was 0.05, telling us dietary beta carotene accounted for just 5 per cent of plasma beta carotene. Adding these other variables in, our new R-squared value is 0.099, telling us this new model predicts about 10 per cent of beta carotene, which still isn't much, but these new variables account for an additional 5 per cent of the variance in plasma beta carotene.

We can also calculate our beta estimates to estimate the number of standard deviations by which the outcome will change as a result of one standard deviation change in each predictor variable.

```
> lm.beta(fit2)

      betadiet      fat      fiber      retdiet
0.14450791 -0.15552929  0.21723695 -0.03533097

> sd(betadiet) * 0.1445

[1] 213.238

> sd(fat) * 0.1555

[1] 5.212567

> sd(fiber) * 0.2172

[1] 1.159421

> sd(retdiet) * 0.0353

[1] 20.80804
```

### 3.1 Bonus: A Shortcut

We've learned to do all of this by hand, but there's a bit of a shortcut: the R Commander package (Rcmdr). When you launch it, it will open a GUI window that will allow you to do most of the things we just did, and more.

## 4 Some Additional Information

### 4.1 Importing Files From Different Statistical Software

The R package 'foreign' is useful for importing datasets that are formatted for different statistical software programs. The foreign package can read in files stored for Minitab, SAS, SPSS, Stat, and more. For more information, see <https://cran.r-project.org/web/packages/foreign/foreign.pdf>.

### 4.2 Additional Resources

#### Websites

- UCLA's Institute for Digital Research and Education has some excellent resources for learning about R in general and statistical analysis in particular: <http://www.ats.ucla.edu/stat/r/>.

- Quick-R has several brief vignettes covering various aspects of R, including basic and advanced statistics and graphs: <http://www.statmethods.net/index.html>
- Dr. William B. King of Coastal Carolina University has a nice set of tutorials on statistics in R: <http://ww2.coastal.edu/kingw/statistics/R-tutorials/>

## **Books**

There are many, many books on statistics in R. These are a few I've found useful.

- Crawley, Michael J. Statistics: An Introduction Using R. Wiley, 2014.
- Field, Andy and Jeremy Miles. Discovering Statistics Using R. SAGE Publications, 2012.
- Lewis, N.D. 100 Statistical Tests in R. Heather Hills Press, 2013.